UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/975,955	10/15/2001	David A. Baldwin	58032.000006	9673

7590 11/28/2005

Ensoport Internetworks
Suite 300
2401 Pennsylvania Ave, NW
Washington, DC 20037

EXAMINER	
ZHONG, CHAD	
ART UNIT	PAPER NUMBER
2152	

DATE MAILED: 11/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/975,955	BALDWIN, DAVID A.	
	Examiner	Art Unit	
	Chad Zhong	2152	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 January 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-3 are presented for examination.

2. It is noted that although the present application does contain line numbers in specification and claims, the line numbers in the claims do not correspond to the preferred format. The preferred format is to number each line of every claim, with each claim beginning with line 1. For ease of reference by both the Examiner and Applicant all future correspondence should include the recommended line numbering.

3. The use of the trademark ensoRAIS among others have been noted in this application (pg 7, line 27). It should be capitalized wherever it appears and be accompanied by the generic terminology. Appropriate correction is required.

Objections

4. An examination of this application reveals that applicant is unfamiliar with patent prosecution procedure. While an inventor may prosecute the application, lack of skill in this field usually acts as a liability in affording the maximum protection for the invention disclosed. Applicant is advised to secure the services of a registered patent attorney or agent to prosecute the application, since the value of a patent is largely dependent upon skilled preparation and prosecution. The Office cannot aid in selecting an attorney or agent.

A listing of registered patent attorneys and agents is available on the USPTO Internet web site http://www.uspto.gov in the Site Index under "Attorney and Agent Roster." Applicants may also obtain a list of registered patent attorneys and agents located in their area by writing to the Mail Stop OED, Director of the U. S. Patent and Trademark Office, PO Box 1450, Alexandria, VA 22313-1450

5 Claim 3 is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. Claim 3 is objected to as a improper multiple dependent claim, See also MPEP § 608.01(n), "Infringement Test" for dependent claims. The test for a proper dependent claim is whether the dependent claim includes every limitation of the parent claim. The test is not whether the claims differ in scope. A proper dependent claim shall not conceivably be infringed by anything which would not also infringe the basic claim. Specifically, Claim 3 does not refer back in the alternative only, i.e. Claim 5. A gadget according to claim 3 and 4, further comprising --- is an example of improper multiple dependent claim.

6. Claim 3 is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. Specifically, "will be" is based on presumption rather than further limiting the claims, thus, the claim is improper dependent.

7. Figure 1-3 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

7. The listing of references in the specification is not a proper information disclosure statement. 37 CFR 1.98(b) requires a list of all patents, publications, or other information submitted for consideration by the Office, and MPEP § 609.04(a) states, "the list may not be incorporated into the specification but must be submitted in a separate paper." Therefore, unless the references have been cited by the examiner on form PTO-892, they have not been considered.

Specifically, references

US 6,122,756, US 5,862,312, US 5,202,980, US 6,128,277, US 5,361,347, US 5,841,775, US 6,253,230, US 6,141,759

are not currently submitted as part of the IDS

8. Claim 1 is objected to because of the following informalities: parenthesis with further defining claim elements, i.e. (memory, CPU, disk). Appropriate correction is required.

9. Applicant is required to update the status (pending, allowed, etc.) of all parent priority applications in the first line of the specification. The status of all citations of US filed applications in the specification should also be updated where appropriate.

10. A substitute specification excluding the claims is required pursuant to 37 CFR 1.125(a) because portion of the specification is missing, the specification stopped after page 9, additional information is needed to have proper support for the claims.

A substitute specification must not contain new matter. The substitute specification must be submitted with markings showing all the changes relative to the immediate prior version of the specification of record. The text of any added subject matter must be shown by underlining the added text. The text of any deleted matter must be shown by strike-through except that double brackets placed

before and after the deleted characters may be used to show deletion of five or fewer consecutive characters. The text of any deleted subject matter must be shown by being placed within double brackets if strike-through cannot be easily perceived. An accompanying clean version (without markings) and a statement that the substitute specification contains no new matter must also be supplied. Numbering the paragraphs of the specification of record is not considered a change that must be shown.

Claim Rejections - 35 USC § 112

11. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

12. Claims 1-3 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

There is no where in the specification that teaches how to do claim 1 and 2, as well as claim 3 is in consequence of claims 1 and 2, appropriate correction to the specification is required to overcome the enablement rejection.

13. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

14. Claims 1-3 are rejected as failing to define the invention in the manner required by 35 U.S.C. 112, second paragraph.

The claim(s) are narrative in form and replete with indefinite and functional or operational language. The structure which goes to make up the device must be clearly and positively specified. The structure must be organized and correlated in such a manner as to present a complete operative device. The claim(s) must be in one sentence form only. Note the format of the claims in the patent(s) cited.

15. Claim 3 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

16. The term "significantly" in claim 3 is a relative term which renders the claim indefinite. The term "significantly" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

17. Claims 1, 2 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. The following terms lack antecedent basis:

   i. the exact software image - claim 1, line 13.

   ii. the least loaded - claim 2, line 26.

   iii. the hardware devices - claim 2, line 32.

b. The claim language in the following claims is not clearly understood, rendering the claims indefinite:

   i. As per claim 2, line 31-32, it is not clearly understood whether "a full level of quality assurance testing" refers to "a full level of quality assurance testing" in claim 2, lines 29-30 (i.e. if they are the same, the word such as "said" or "the" must be used);

Claim Analysis

19 The examiner will interpret claim 1 as follows, configuration of VLAN to have plurality of service zones, each zone having separate traffic flow; router access list to provide protection across the VLAN traffic mentioned above; a plurality of redundant inexpensive server machines with same configurations, storing software images of other servers and share said images with other machines on the network.

Claim Rejections - 35 USC § 103

20. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

21. Claims 1-3 are rejected under 35 U.S.C. 103(a) as being unpatentable over Aziz, US 2003/0154279, in view of Aziz (hereinafter Aziz_b), US 6,779,016, in further view of 'Ten Minute LDAP Tutorial', Orelly, 2001 (hereinafter Orelly).

22. As per claim 1, Aziz teaches:

a method for creating a well-defined architecture that combines configurations at the network, server and storage tier of an infrastructure in order to provide for scalability of services incrementally, increased performance, and enhanced security made up of the following configuration tasks:

(a) Configuring several Virtual Local Area Network (VLAN) segments in order to separate traffic from server to disk, from content switch to server, from user or WAN connected Ethernet segment to content switch (pg 4-5, [0082-0083], wherein database 131 contains VLAN configurations, and messages instruct the hardwdare abstraction layer how to place CPUs of the computing grid in particular VLANs, another words, different network hardware are assigned to different VLANs);

(b) Configuration of router access lists such that traffic is protected across the above mentioned VLAN segments ([0025], [0082], where the gateway routes information storage and retrieval requests; Table 8A, 8B, wherein the router access list is controlled by firewall, furthermore, routing information protocol is one of plurality of protocols configurable by the firewall).

(c) Configuration of many redundant inexpensive server machines ([0089-0094], adding/removing servers from lists, load balancing the servers, wherein the IDCs can be configured in similar fashion with each other, user can define same amount/type of computer elements, firewalls, load balancers etc.);

(d) Above mentioned machines are configured exactly the same (memory, CPU, disk) ([0087]; [0094], instant data centers can be created with the use of the editor, another words, hardware devices with identical configuration can be grouped together within the same load balancing server tier; a blueprint ('DNA') for creating any number of other IDCs that have the same structure);

(e) Each server contains the exact software image of all other servers and machine dependent configurations are stored in database (pg 7, [0106-0108], service provider 126 loads software image onto any particular IDC for access);

(f) Configuration of Network File System (NFS) such that machines can share storage and file locking is managed via NFS ([0224], table 8B, 8C, wherein the storage and the images within the storage are accessible with the proper authentication rights, thus the NFS performs the sharing and the locking aspects);

(g) This storage is configured in a Redundant Array of Inexpensive Disk (RAID) configuration ([0271]).

Aziz does not explicitly teach (a) storing of images and configurations within LDAP, (b) Network Attached Storage (NAS) technology

However, Orelly teaches storing of images and configuration within the LDAP server (Orelly, Fig B-1, wherein each entry is unique and directories are independent). It would have been obvious to one of

ordinary skill in this art at the time of invention was made to incorporate Orelly with Aziz because the combination would improve the scalability for Aziz's system by running directly over existing TCP/IP and SSL protocols, moreover, it would improve the uniqueness of each entry, allowing for multiple independent entries easily identifiable upon retrieval. Aziz_b discloses NAS technology (Aziz_b, Col. 3, lines 8-12, lines 18-24; Col. 12, lines 50-55; Col. 7, lines 60-67; Col. 6, lines 1-13, for the advantages of compatibility). It would have been obvious to one of ordinary skill in this art at the time of invention was made to incorporate Aziz_b with Aziz because the combination would improve the compatibility and security for Aziz's system by extending the sharing of resources over the communications network (Aziz_b, Col. 12, lines 50-57).

23. As per claim 2, Aziz, Orelly, Aziz_b disclose the invention substantially as rejected in claim 1 above, including:

(a) Each server mentioned in claim 1 will have an exact copy of the complete software grouping (Aziz, [0094-0095]; [0106-0108]);

(b) The software grouping consists of an Email MTA, Web-based email front-end, POP daemon, Chat daemon, Web daemon, backup server software, monitoring daemon and agents, Web-based content portal, and additional software as it becomes useful to users of service providerships (Aziz, [0110], [0111], table 8b, [0271], [0274], [0110]);

(c) Users of the system will be directed to the least loaded and most available server by way of a content switch (Aziz, [0183], [0260], [0237], load balancer achieves this purpose);

(d) Any server will be able to handle the user request for service or software application (Aziz, [0107]; [0260], [0237], in a distributed load balancing system, any server within the load balancing group is able to handle client requests);

(e) Software can be added to the grouping at any time after it has been through a full level of quality assurance testing (Aziz, [0095]; [0101]; [0115], wherein the new software modules can be added to the

servers after a test such as the stress testing);

(f) After new software, bug fixes or security updates have been through a full level of quality assurance testing, they can automatically be pushed to the hardware devices within the architecture defined by claim 1 (Aziz, [0095]).

Aziz does not explicitly teach:

(b) IMAP daemon

(g) A Lightweight Directory Access Protocol (LDAP) configuration database will store any independent server configurations that will identify slight differences in the software.

(h) Above mentioned LDAP configuration database will not impact the software grouping at all, but will serve to extend the grouping.

However, Official Notice is taken (see MPEP 2144.03) IMAP daemon is well known and routinely used for electronic mail purposes at the time of the invention was made. It would have been obvious to the person of ordinary skill in the art at the time of the invention to have included IMAP daemon with Aziz because doing so would improve the capability of Aziz by allowing for another form of electronic mail protocol to work with the POP protocol, furthermore, Using IMAP an email client program can not only retrieve email but can also manipulate message stored on the server, without having to actually retrieve the messages. So messages can be deleted, have their status changed, multiple mail boxes can be managed.

Orelly teaches:

(g) A Lightweight Directory Access Protocol (LDAP) configuration database will store any independent server configurations that will identify slight differences in the software (Orelly, Fig B-1, wherein the global naming model ensures unique entries within the LDAP).

(h) Above mentioned LDAP configuration database will not impact the software grouping at all, but will serve to extend the grouping (Orelly, Fig B-1, as names and attributes are added to the LDAP

database, there would not be any conflicts because the entries are unique and scalability would be improved as well as the database support increases to additional entries).

24. As per claim 3, Aziz, Orelly, Aziz_b disclose the invention substantially as rejected in claim 1 above, including:

the methods in claims 1 and 2 will significantly decrease cost, increase scalability, increase redundancy and enhance security (Aziz, [0077]; [0095]; [0105]).

Conclusion

22. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following patents and publications are cited to further show the state of the art with respect to "ENSOBOX CLUSTERED SERVICES ARCHITECTURE: TECHNIQUES FOR ENABLING THE CREATION OF SCALABLE, ROBUST, AND INDUSTRIAL STRENGTH INTERNET SERVICES PROVIDER APPLIANCE".

i.	US 2002/0103889	Markson et al.
ii.	US 6421711	Blumenau; Steven M. et al.
iii.	US 6212559	Bixler et al.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chad Zhong whose telephone number is (571)272-3946. The examiner can normally be reached on M-F 7:15 to 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JAROENCHONWANIT, BUNJOB can be reached on (571)272-3913. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CZ
November 21, 2005

BUNJOB JAROENCHONWANIT
PRIMARY EXAMINER

Notice of References Cited	Application/Control No. 09/975,955	Applicant(s)/Patent Under Reexamination BALDWIN, DAVID A.	
	Examiner Chad Zhong	Art Unit 2152	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-2003/0154279	08-2003	Aziz, Ashar	709/225
	B	US-6,779,016	08-2004	Aziz et al.	709/201
	C	US-2002/0103889	08-2002	Markson et al.	709/223
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	"The Ten Minute LDAP Tutorial", Orelly, 2001
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Appendix B. The Ten-Minute LDAP Tutorial

Contents:

The Lightweight Directory Access Protocol (LDAP) is one of the pre-eminent directory services deployed in the world today. Over time, system administrators are likely to find themselves dealing with LDAP servers and clients in a number of contexts. This tutorial will give you an introduction to the LDAP nomenclature and concepts you'll need when using the material in Chapter 6, "Directory Services".

The action in LDAP takes place around a data structure known as an *entry*. Figure B-1 is a picture to keep in mind as we look at an entry's component parts.

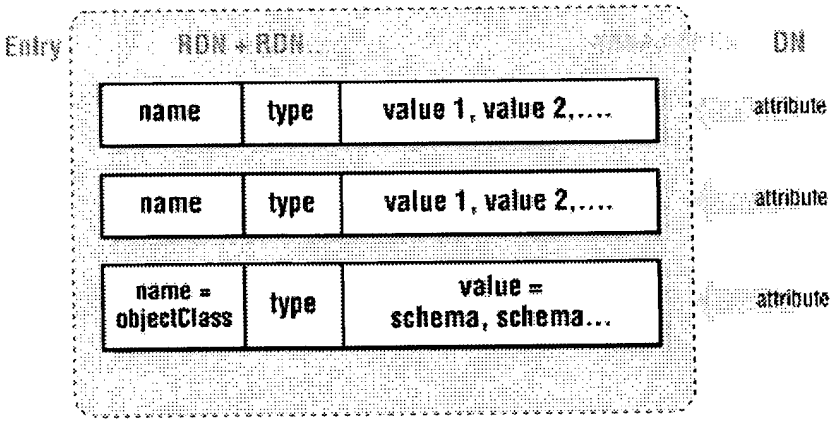


Figure B-1. The LDAP entry data structure

An entry has a set of named component parts called *attributes* that hold the data for that entry. To use database terms, they are like the fields in a database record. In Chapter 6, "Directory Services" we'll use Perl to keep a list of machines in an LDAP directory. Each machine entry will have attributes like name, model, location, owner, etc.

Besides its name, an attribute consists of a *type* and a set of *values* that conform to that type. If you are

storing employee information, your entry might have a `phone` attribute that has a type of `telephoneNumber`. The values of this attribute might be that employee's phone numbers. A type also has a *syntax* that dictates what kind of data can be used (strings, numbers, etc.), how it is sorted, and how it is used in a search (is it case-sensitive?).

Each entry has a special attribute called `objectClass`. `objectClass` contains multiple values that, when combined with server and user settings, dictate which attributes must and may exist in that particular entry.

Let's look a little closer at the `objectClass` attribute for a moment because it illustrates some of the important qualities of LDAP and allows us to pick off the rest of the jargon we haven't seen yet. If we consider the `objectClass` attribute, we notice the following:

LDAP is object-oriented

Each of the values of an `objectClass` attribute is a name of an object class. These classes either define the set of attributes that can or must be in an entry, or expand on the definitions inherited from another class.

Here's an example: an `objectClass` in an entry may contain the string `residentialPerson`. RFC2256, which has the daunting title of "A Summary of the X.500(96) User Schema for use with LDAPv3," defines the `residentialPerson` object class like this:

```
residentialPerson
    ( 2.5.6.10 NAME 'residentialPerson' SUP person STRUCTURAL MUST l
      MAY ( businessCategory $ x121Address $ registeredAddress $
      destinationIndicator $ preferredDeliveryMethod $ telexNumber $
      teletexTerminalIdentifier $ telephoneNumber $
      internationaliSDNNumber $
      facsimileTelephoneNumber $ preferredDeliveryMethod $ street $
      postOfficeBox $ postalCode $ postalAddress $
      physicalDeliveryOfficeName $ st $ l ) )
```

This definition says that an entry of object class `residentialPerson` must have an `l` attribute (short for locality) and may have a whole other set of attributes (`registeredAddress`, `postOfficeBox`, etc.). The key part of the specification is the `SUP person` string. It says that the superior class (the one that `residentialPerson` inherits *its* attributes from) is the `person` object class. That definition looks like this:

```
person
    ( 2.5.6.6 NAME 'person' SUP top STRUCTURAL MUST ( sn $ cn )
      MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

So an entry with object class `residentialPerson` must have `sn` (surname), `cn` (common name), and `l` (locality) attributes and may have the other attributes listed in the `MAY` sections of these two RFC excerpts. We also know that `person` is the top of the object hierarchy for `residentialPerson` since its superior class is the special abstract class `top`.

In most cases, you can get away with using the pre-defined standard object classes. If you need to construct entries with attributes not found in an existing object class, it is usually good form to locate the closest existing object class and build upon it, like `residentialPerson`, builds upon

`person` above.

LDAP has its origins in the database world

A second quality we see in `objectClass` is LDAP's database roots. A collection of object classes that specify attributes for the entries in an LDAP server is called a *schema*. The RFC we quoted above is one example of an LDAP schema specification. We won't be addressing the considerable issues surrounding schema in this book. Like database design, schema design can be a book topic in itself, but you should at least be familiar with the term "schema" because it will pop up later.

LDAP is not limited to storing information in strict tree structures

One final note about `objectClass` to help us move from our examination of a single entry to the larger picture: our previous object class example specified `top` at the top of the object hierarchy, but there's another quasi-superclass worth mentioning: `alias`. If `alias` is specified, then this entry is actually an alias for another entry (specified by the `aliasedObjectName` attribute in that entry). LDAP strongly encourages hierarchical tree structures, but it doesn't demand them. It's important to keep this flexibility in mind when you code to avoid making incorrect assumptions about the data hierarchy on a server.

B.1. LDAP Data Organization

So far we've been focused on a single entry, but there's very little call for a directory that contains only one entry. When we expand our focus and consider a directory populated with many entries, we are immediately faced with the question that began this chapter: How do you find anything?

The stuff we've discussed so far all falls under what the LDAP specification calls its "information model." This is the part that sets the rules for how information is represented. But for the answer to our question we need to look to LDAP's "naming model," which dictates how information is organized.

If you look at Figure B-1, you can see we've discussed all of the parts of an entry except for its name. Each entry has a name, known as its *Distinguished Name* (DN). The DN consists of a string of *Relative Distinguished Name*s (RDNs). We'll return to DNs in a moment, but first let's concentrate on the RDN building blocks.

An RDN is composed of one or several attribute name-value pairs. For example: `cn=JaySekora` (where `cn` stands for "common name") could be an RDN. The attribute name is `cn` and the value is `Jay Sekora`.

Neither the LDAP nor the X.500 specifications dictate which attributes should be used to form an RDN. They do require RDNs to be unique at each level in a directory hierarchy. This restriction exists because LDAP has no inherent notion of "the third entry in the fourth branch of a directory tree" so it must rely on unique names at each level to distinguish between individual entries at that level. Let's see how this restriction plays out in practice.

Take, for instance, another example RDN: `cn=Robert Smith`. This is probably not a good RDN choice, since there is likely to be more than one Robert Smith in an organization of even moderate size. If you have a large number of people in your organization and your LDAP hierarchy is relatively flat, name collisions like this are to be expected. A better entry would combine two attributes, perhaps `cn=Robert Smith+l=Boston`. (Attributes in RDNs are combined with a plus sign.)

Our revised RDN, which appends a locality attribute, still has problems. We may have postponed a name clash, but we haven't eliminated the possibility. Furthermore, if Smith moves to some other facility, we'll have to change both the RDN for the entry *and* the location attribute in the entry. Perhaps the best RDN we could use would be one with a unique and immutable user ID for this person. For example, we could use that person's email address so the RDN would be `uid=rsmith`. This example should give you a taste of the decisions involved in the world of schemas.

Astute readers will notice that we're not really expanding our focus; we're still puttering around with a single entry. The RDN discussion was a prelude to this; here's the real jump: entries live in a tree-like[1] structure known as a *Directory Information Tree* (DIT) or just *directory tree*. The latter is probably the preferred term to use, because in X.500 nomenclature DIT usually refers to a single universal tree, similar to the global DNS hierarchy or the Management Information Base (MIB) we'll be seeing later when we discuss SNMP.

> [1]It is called *tree-like* rather than just *tree* because the `alias` object class we mentioned earlier allows you create a directory structure that is not strictly a tree (at least from a computer-science, directed-acyclic-graph perspective).

Let's bring DNs back into the picture. Each entry in a directory tree can be located by its Distinguished Name. A DN is composed of an entry's RDN followed by all of the RDNs (separated by commas or semi-colons) found as you walk your way back up the tree towards the root entry. If we follow the arrows in Figure B-2 and accumulate RDNs as we go, we'll construct DNs for each highlighted entry.

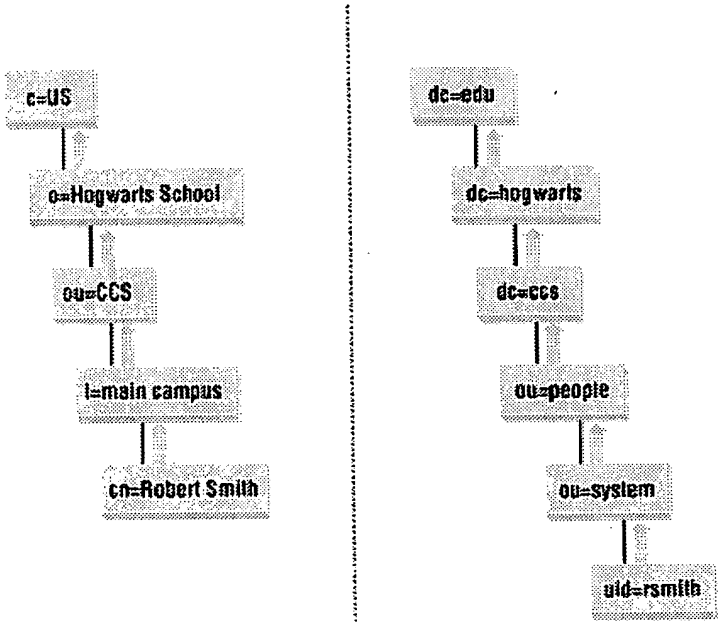


Figure B-2. Walking back up the tree to produce a DN

In the first picture, our DN would be:

```
cn=Robert Smith, l=main campus, ou=CCS, o=Hogwarts School, c=US
```

In the second, it is:

```
uid=rsmith, ou=systems, ou=people, dc=ccs, dc=hogwarts, dc=edu
```

ou is short for organizational unit, o is short for organization, dc stands for "domain component" à la DNS, and c is for country (Sesame Street notwithstanding).

An analogy is often made between DNs and absolute pathnames in a filesystem, but DNs are more like postal addresses because they have a "most specific component first" ordering. In a postal address like:

Pat Hinds

288 St. Bucky Avenue

Anywhere, MA 02104

USA

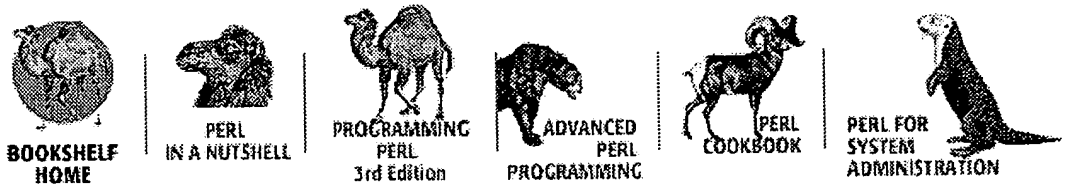
you start off with the most specific object (the person) and get more vague from there, eventually winding up at the least specific component (the country or planet). So too it goes with DNs. You can see this ordering in our DN examples.

The very top of the directory tree is known as the directory's *suffix*, since it is the end portion of every DN in that directory tree. Suffixes are important when constructing a hierarchical infrastructure using multiple delegated LDAP servers. Using an LDAPv3 concept known as a *referral*, it is possible to place an entry in the directory tree that essentially says, "for all entries with this suffix, go ask that server instead." Referrals are specified using an *LDAP URL*, which look similar to your run-of-the-mill web URL except they reference a particular DN or other LDAP-specific information. Here's an example from RFC2255, the RFC that specifies the LDAP URL format:

```
ldap://ldap.itd.umich.edu/o=University%20of%20Michigan,c=US?postalAddress
```

PREVIOUS | HOME | NEXT

A.1. References for More Information | BOOK INDEX | C. The Eight-Minute XML Tutorial



Advantages of LDAP

- Global naming model ensures unique entries
- Allows for multiple independent directories
- Extensible to meet future/local requirements
- Runs directly over TCP/IP and SSL
- Has broad industry support
- Based on existing deployed technologies

Previous slide Next slide Back to first slide View graphic version

Organization TC2100 Bldg./Room RANDOLPH

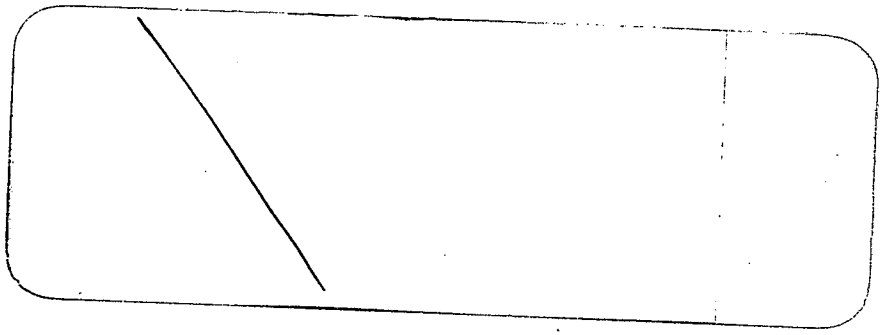
U. S. DEPARTMENT OF COMMERCE
COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
IF UNDELIVERABLE RETURN IN TEN DAYS

OFFICIAL BUSINESS

UNITED STATES POS
02 1A
000420447
MAILED FRO

AN EQUAL OPPORTUNITY EMPLOYER

No longer at this address Please return to sender



UNITED AUK
3751
S

RECEIVED
JAN 27 2006
USPTO MAIL CENTER